Advanced Systems Lab Report
Autumn Semester 2018

Name: Nicolas Küchler

Legi: 14-712-129

Grading

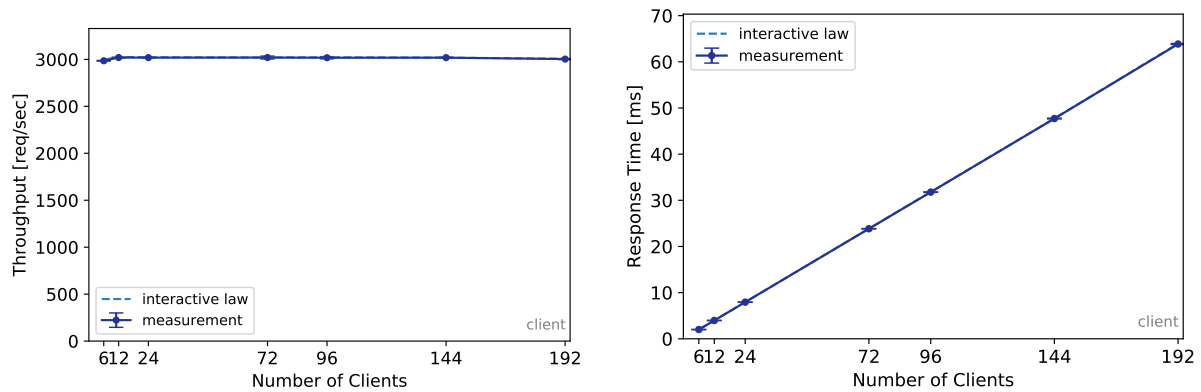| Section	Points
1	
2	
3	
4	
5	
6	
7	
Total	

Figure 1: Throughput and response time with interactive law in read-only workload

1 System Overview (75 pts)

Describe the implementation of your system and highlight design decisions relevant for the experiments. Explain how messages are parsed and how statistics are gathered in a multi-threaded setting. Provide figures containing all the threads and queues in your system (including the network and the memcached servers). Include illustrations that show how requests of different types are handled (e.g., components involved in processing the request and method calls). Please include all details necessary to understand artifacts and effects in your experiments that arise from your implementation choices.

2 Baseline without Middleware (75 pts)

In this experiments you study the performance characteristics of the memtier clients and memcached servers.

2.1 One Server

Both, for a read-only and write-only workload plot the throughput and the response time as a function of NumClients. All clients are connected to a single memcached instance.

Use 3 load generating VMs, with one memtier (CT=2) each, and vary the number of virtual clients (VC) per memtier thread between 1 and 32. Show how the behavior of the server changes as we add more clients.

Number of servers	1
Number of client machines	3
Instances of memtier per machine	1
Threads per memtier instance	2
Virtual clients per thread	[1..32]
Workload	Write-only and Read-only
Multi-Get behavior	N/A
Multi-Get size	N/A
Number of middlewares	N/A
Worker threads per middleware	N/A
Repetitions	3 or more

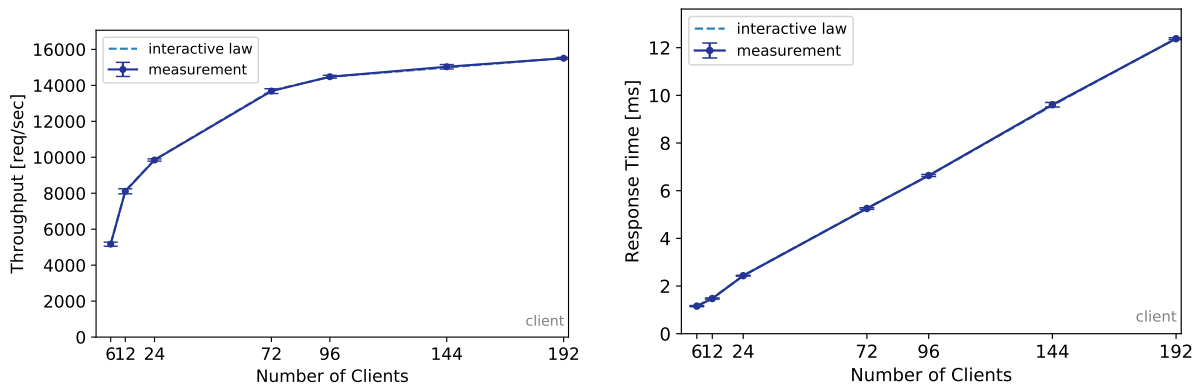need better reference to the experiment

Figure 2: Throughput and response time with interactive law in write-only workload

2.1.1 Explanation

Describe in which phase the memcached servers are under-saturated, saturated, or over-saturated. Describe how throughput and response time correlate. Explain what further conclusions can be drawn from the experiment.

2.2 Two Servers

For a read-only and write-only workload plot throughput and response time as a function of NumClients. The clients are connected to two memcached instances.

Use 1 load generating VM, with one memtier (CT=1) connected to each memcached instance (two memcache instances in total), and vary the number of virtual clients (VC) per memtier thread between 1 and 32. Show how the behavior of the server changes and explain what conclusions we can draw from this experiment.

Number of servers	2
Number of client machines	1
Instances of memtier per machine	2
Threads per memtier instance	1
Virtual clients per thread	[1..32]
Workload	Write-only and Read-only
Multi-Get behavior	N/A
Multi-Get size	N/A
Number of middlewares	N/A
Worker threads per middleware	N/A
Repetitions	3 or more (at least 1 minute each)

need better reference to the experiment

2.2.1 Explanation

Describe how this experiment compares to the previous section. Which results are the same and which ones differ? Explain what further conclusions can be drawn from the experiment.

2.3 Summary

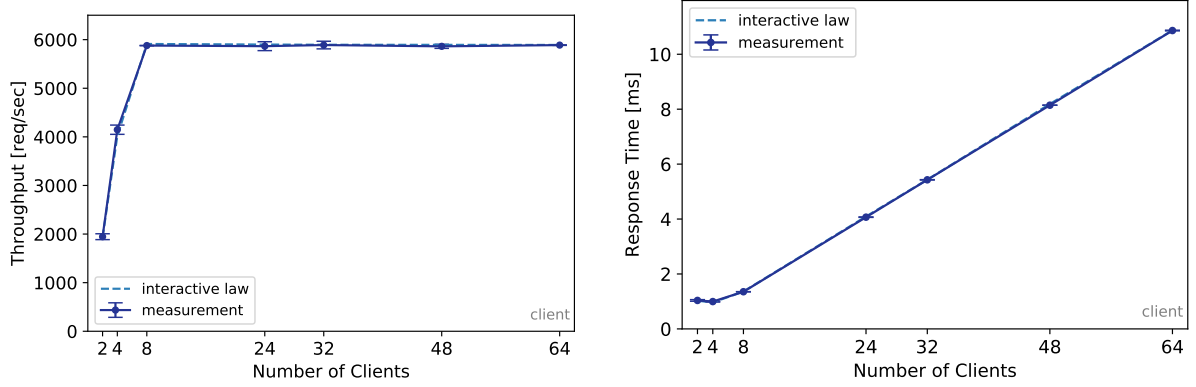Based on the experiments above, fill out the following table:

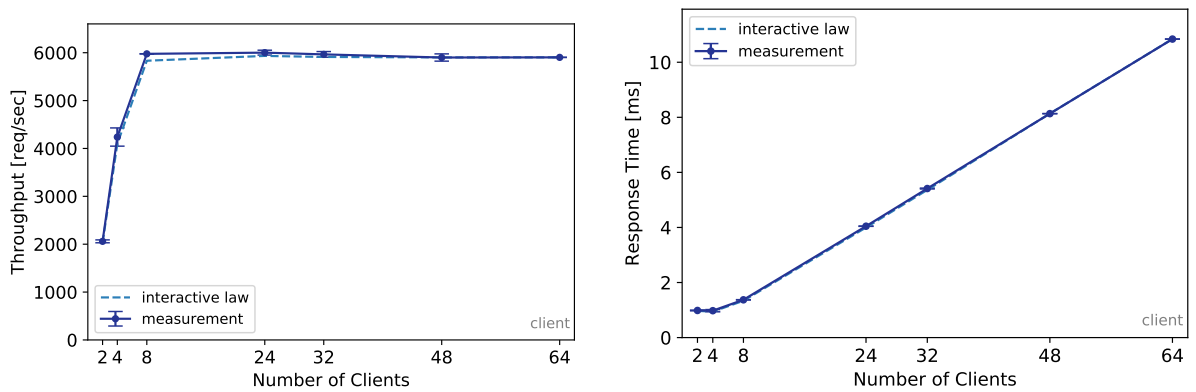Figure 3: Throughput and response time with interactive law in read-only workload

Figure 4: Throughput and response time with interactive law in write-only workload

Maximum throughput of different VMs.

	Read-only workload	Write-only workload	Configuration gives max. throughput
One memcached server			
One load generating VM			

Write at least two paragraphs about how both results relate. Describe what is the bottleneck of this setup is. If the maximum throughput for both experiments is the same, explain why. If it is not the case, explain why not. Write down key take-away messages about the behaviour of the memtier clients and the memcached servers.

3   Baseline with Middleware (90 pts)

In this set of experiments, you will have to use 1 load generator VM and 1 memcached server, measuring how the throughput of the system changes when increasing the number of clients. Scaling virtual clients inside memtier has to be done as explained in the previous sections. Plot both throughput and response time as measured on the middleware.

3.1   One Middleware

Connect one load generator machine (one instance of memtier with CT=2) to a single middleware and use 1 memcached server. Run a read-only and a write-only workload with increasing number of clients (between 2 and 64) and measure response time *both at the client and at the middleware*, and plot the throughput and response time measured in the middleware.

Repeat this experiment for different number of worker threads inside the middleware: 8, 16, 32, 64.

Number of servers	1
Number of client machines	3
Instances of memtier per machine	1
Threads per memtier instance	2
Virtual clients per thread	[1..32]
Workload	Write-only and Read-only
Multi-Get behavior	N/A
Multi-Get size	N/A
Number of middlewares	1
Worker threads per middleware	[8..64]
Repetitions	3 or more (at least 1 minute each)

need better reference to the experiment

3.1.1   Explanation

Provide a detailed analysis of the results (e.g., bottleneck analysis, component utilizations, average queue lengths, system saturation). Add any additional figures and experiments that help you illustrate your point and support your claims.

3.2   Two Middlewares

Connect one load generator machine (two instances of memtier with CT=1) to two middlewares and use 1 memcached server. Run a read-only and a write-only workload with increasing number of clients (between 2 and 64) and measure response time *both at the client and at the middleware*, and plot the throughput and response time as measured in the middleware.
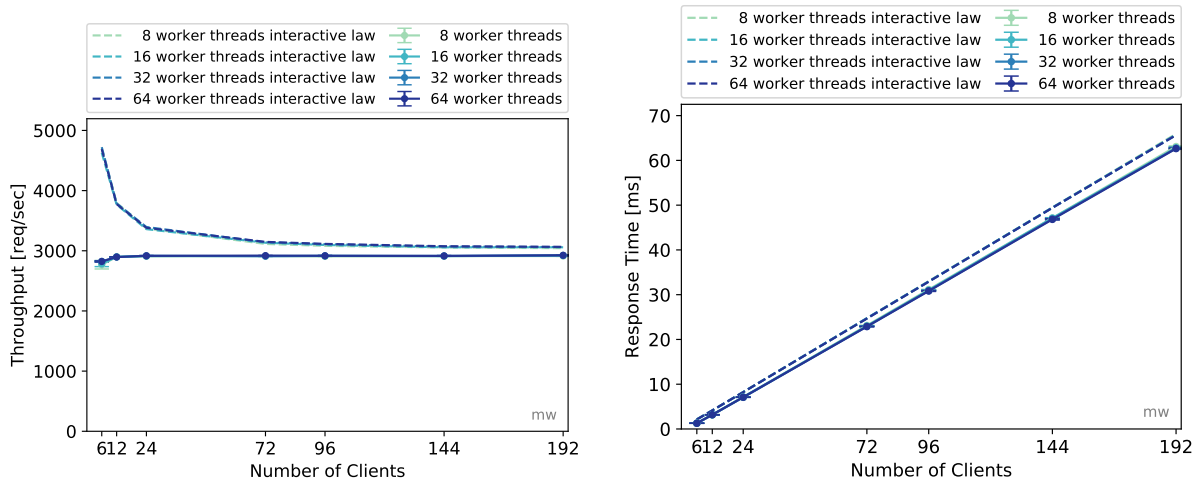
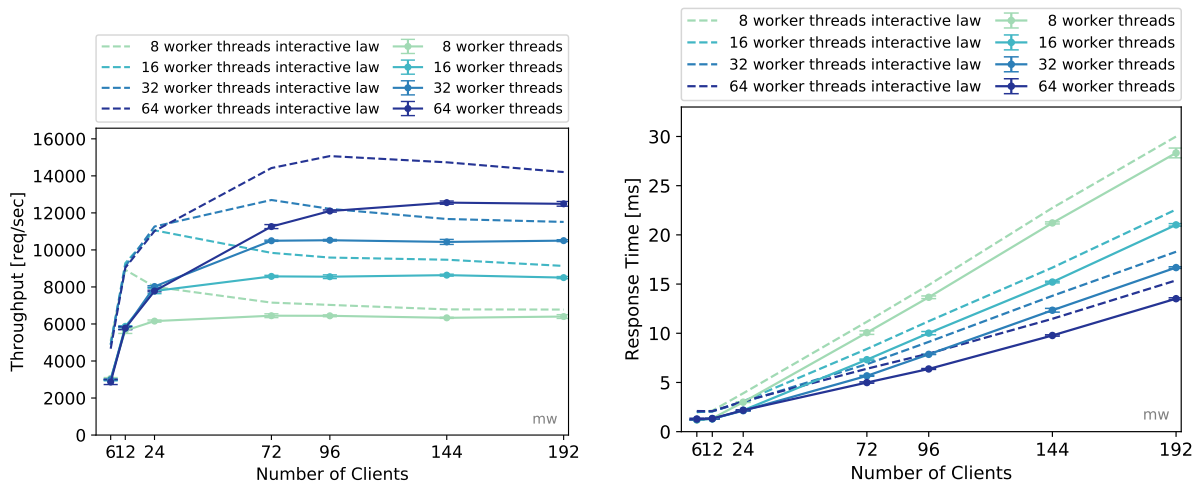Figure 5: Throughput and response time with interactive law in read-only workload

Figure 6: Throughput and response time with interactive law in write-only workload
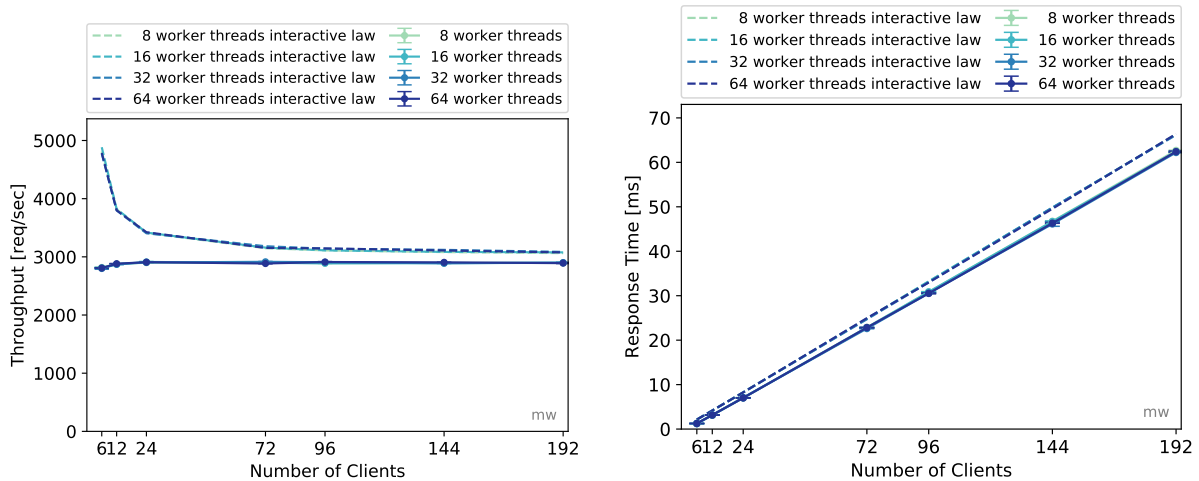
Figure 7: Throughput and response time with interactive law in read-only workload

Repeat this experiment for different number of worker threads inside the middleware: 8, 16, 32, 64.

If in your experiment the middleware is not the bottleneck, repeat the experiment that reaches the highest throughput but using two load generator VMs (each with 2x memtier CT=1) instead of one. Otherwise, explain how you know that the middlewares are the limiting factor in terms of throughput.

Number of servers	1
Number of client machines	3
Instances of memtier per machine	2
Threads per memtier instance	1
Virtual clients per thread	[1..32]
Workload	Write-only and Read-only
Multi-Get behavior	N/A
Multi-Get size	N/A
Number of middlewares	2
Worker threads per middleware	[8..64]
Repetitions	3 or more (at least 1 minute each)

3.2.1 Explanation

Provide a detailed analysis of the results (e.g., bottleneck analysis, component utilizations, average queue lengths, system saturation). Add any additional figures and experiments that help you illustrate your point and support your claims.

3.3 Summary

Based on the experiments above, fill out the following table. For both of them use the numbers from a single experiment to fill out all lines. Miss rate represents the percentage of GET requests that return no data. Time in the queue refers to the time spent in the queue between the net-thread and the worker threads.

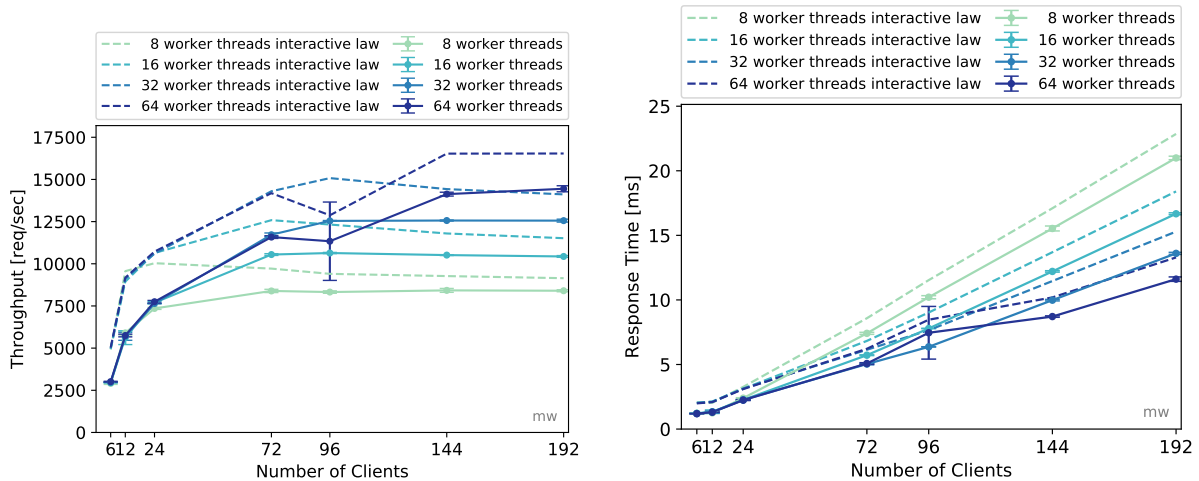need better reference to the experiment

Figure 8: Throughput and response time with interactive law in write-only workload

Maximum throughput for one middleware.

	Throughput	Response time	Average time in queue	Miss rate
Reads: Measured on middleware				
Reads: Measured on clients			n/a	
Writes: Measured on middleware				n/a
Writes: Measured on clients			n/a	n/a

Maximum throughput for two middlewares.

	Throughput	Response time	Average time in queue	Miss rate
Reads: Measured on middleware				
Reads: Measured on clients			n/a	
Writes: Measured on middleware				n/a
Writes: Measured on clients			n/a	n/a

Based on the data provided in these tables, write at least two paragraphs summarizing your findings about the performance of the middleware in the baseline experiments.

4 Throughput for Writes (90 pts)

4.1 Full System

Connect three load generating VMs to two middlewares and three memchached servers. Run a write-only experiment. You need to plot throughput and response time measured on the middleware as a function of number of clients. The measurements have to be performed for 8, 16, 32 and 64 worker threads inside each middleware.
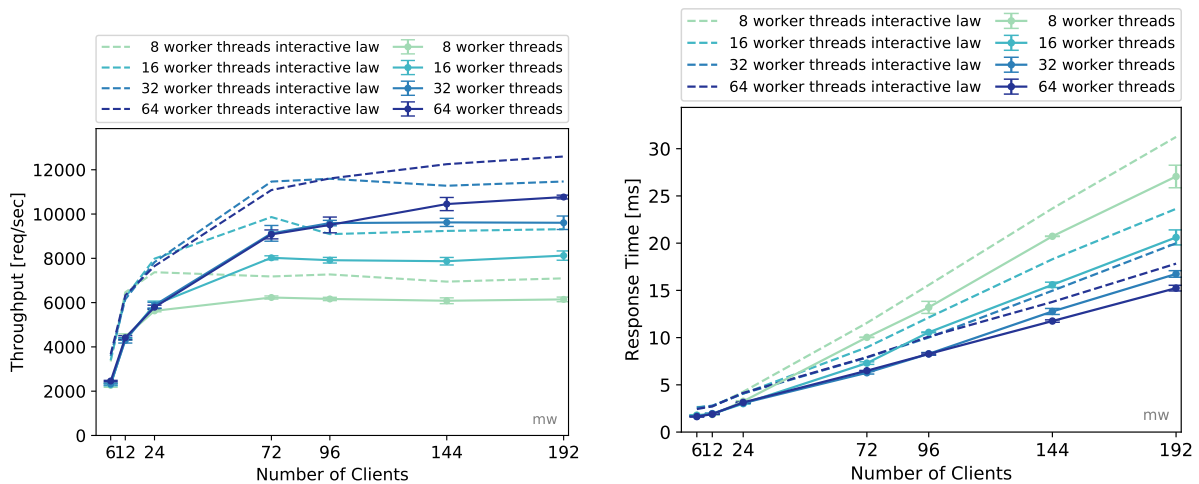
Figure 9: Throughput and response time with interactive law in write-only workload

Number of servers	3
Number of client machines	3
Instances of memtier per machine	2
Threads per memtier instance	1
Virtual clients per thread	[1..32]
Workload	Write-only
Multi-Get behavior	N/A
Multi-Get size	N/A
Number of middlewares	2
Worker threads per middleware	[8..64]
Repetitions	3 or more (at least 1 minute each)

4.1.1 Explanation

Provide a detailed analysis of the results (e.g., bottleneck analysis, component utilizations, average queue lengths, system saturation). Add any additional figures and experiments that help you illustrate your point and support your claims.

> need better reference to experiment

4.2 Summary

Based on the experiments above, fill out the following table with the data corresponding to the maximum throughput point for all four worker-thread scenarios.

Maximum throughput for the full system

	WT=8	WT=16	WT=32	WT=64
Throughput (Middleware)				
Throughput (Derived from MW response time)				
Throughput (Client)				
Average time in queue				
Average length of queue				
Average time waiting for memcached				

Based on the data provided in these tables, draw conclusions on the state of your system for a variable number of worker threads.
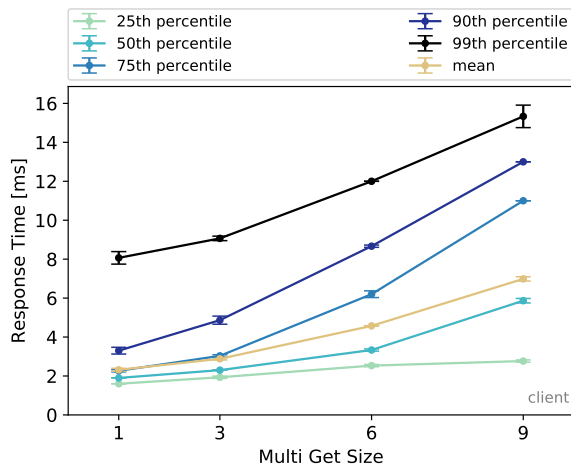
Figure 10: Average response time and 25th, 50th, 75th, 90th and 99th percentiles of different multi-get sizes in sharded mode

5 Gets and Multi-gets (90 pts)

For this set of experiments you will use three load generating machines, two middlewares and three memcached servers. Each memtier instance should have 2 virtual clients in total and the number of middleware worker threads is 64, or the one that provides the highest throughput in your system (whichever number of threads is smaller).

For multi-GET workloads, memtier will generate a mixture of SETs, GETs, and multi-GETs. Memtier only allows to specify the maximum number of keys in a multi-GET request. Therefore, be aware that requests can also contain fewer keys than the provided value. It is recommended to record the average size of the multi-GETs. You will have to measure response time on the client as a function of multi-get size, with and without sharding on the middlewares.

5.1 Sharded Case

Run multi-gets with 1, 3, 6 and 9 keys (memtier configuration) with sharding enabled (multi-gets are broken up into smaller multi-gets and spread across servers). Plot average response time as measured on the client, as well as the 25th, 50th, 75th, 90th and 99th percentiles.

Number of servers	3
Number of client machines	3
Instances of memtier per machine	2
Threads per memtier instance	1
Virtual clients per thread	2
Workload	ratio=1:<Multi-Get size>
Multi-Get behavior	Sharded
Multi-Get size	[1..9]
Number of middlewares	2
Worker threads per middleware	max. throughput config.
Repetitions	3 or more (at least 1 minute each)

> better reference to experiment

5.1.1 Explanation

Provide a detailed analysis of the results (e.g., bottleneck analysis, component utilizations, average queue lengths, system saturation). Add any additional figures and experiments that help you illustrate your point and support your claims.

Figure 11: Average response time and 25th, 50th, 75th, 90th and 99th percentiles of different multi-get sizes in non-sharded mode

5.2   Non-sharded Case

Run multi-gets with 1, 3, 6 and 9 keys (memtier configuration) with sharding disabled. Plot average response time as measured on the client, as well as the 25th, 50th, 75th, 90th and 99th percentiles.

Number of servers	3
Number of client machines	3
Instances of memtier per machine	2
Threads per memtier instance	1
Virtual clients per thread	2
Workload	ratio=1:<Multi-Get size>
Multi-Get behavior	Non-Sharded
Multi-Get size	[1..9]
Number of middlewares	2
Worker threads per middleware	max. throughput config.
Repetitions	3 or more (at least 1 minute each)

5.2.1   Explanation

Provide a detailed analysis of the results (e.g., bottleneck analysis, component utilizations, average queue lengths, system saturation). Add any additional figures and experiments that help you illustrate your point and support your claims.
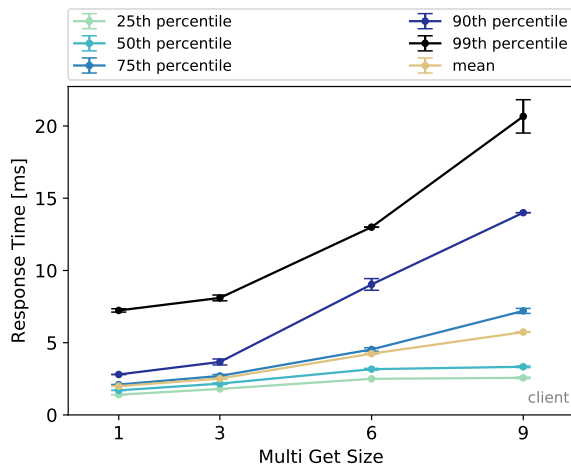
better refer- ence to experi- ment

5.3   Histogram

For the case with 6 keys inside the multi-get, display four histograms representing the sharded and non-sharded response time distribution, both as measured on the client, and inside the middleware. Choose the bucket size in the same way for all four, and such that there are at least 10 buckets on each of the graphs.

5.4   Summary

Provide a detailed comparison of the sharded and non-shareded modes. For which multi-GET size is sharding the preferred option? Provide a detailed analysis of your system. Add any

need better refer- ence to experi- ment
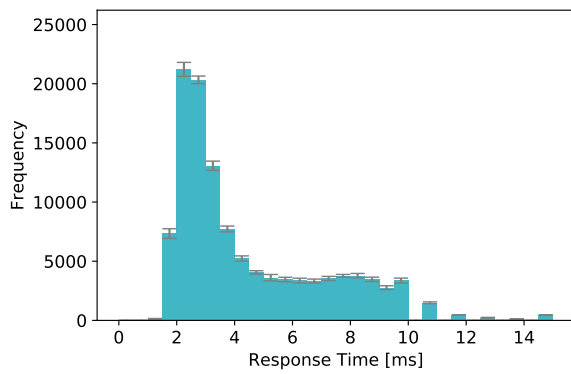
(a) Sharded - Middleware
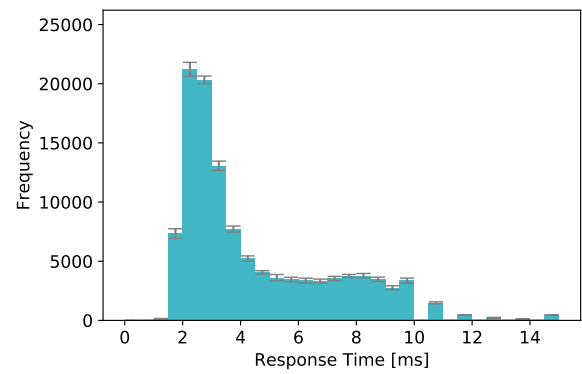
(b) Sharded - Client

(c) Non-Sharded - Middleware

(d) Non-Sharded - Client

Figure 12: Sharded and non-sharded response time distirbution for multi-gets with 6 keys as measured inside the middleware and on the client

Table 1: 2^k3 Experiment Base Table for Write-Only

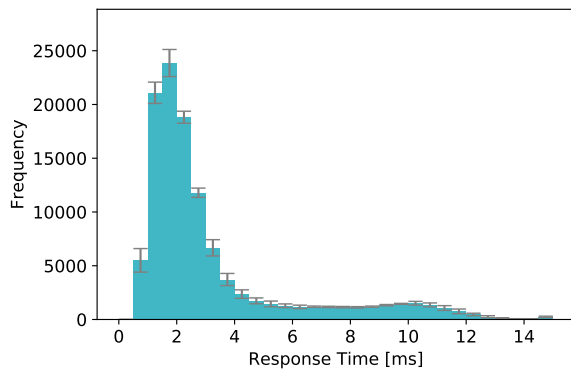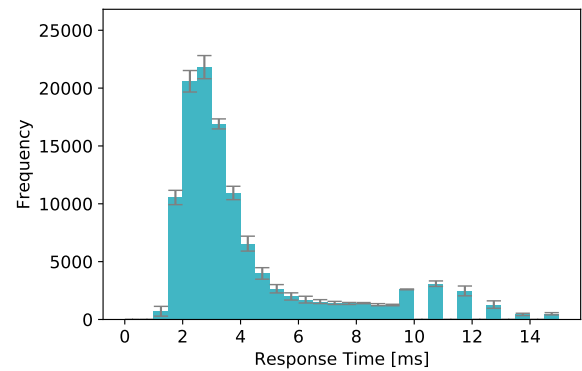| i	I	S	M	W	SM	SW	MW	SMW	Throughput (ops/sec)		Response Time (ms)	
									(y_{i1}, y_{i2}, y_{i3})	$\hat{y_i}$	(y_{i1}, y_{i2}, y_{i3})	$\hat{y_i}$
1	1	-1	-1	-1	1	1	1	-1	(6248, 6306, 6262)	6272	(28.6, 28.5, 28.7)	28.6
2	1	1	-1	-1	-1	-1	1	1	(5293, 5110, 5066)	5157	(33.8, 34.2, 35.5)	34.5
3	1	-1	1	-1	-1	1	-1	1	(8212, 8211, 8210)	8211	(21.4, 21.2, 21.2)	21.3
4	1	1	1	-1	1	-1	-1	-1	(6718, 6585, 6784)	6696	(25.7, 24.7, 25.9)	25.4
5	1	-1	-1	1	1	-1	-1	1	(9474, 9964, 9997)	9812	(16.0, 17.5, 17.4)	17.0
6	1	1	-1	1	-1	1	-1	-1	(7734, 7707, 7639)	7693	(22.2, 22.2, 22.4)	22.3
7	1	-1	1	1	-1	-1	1	-1	(12300, 12250, 11909)	12153	(13.7, 13.8, 13.1)	13.5
8	1	1	1	1	1	1	1	1	(9208, 9864, 9589)	9554	(16.4, 17.1, 16.0)	16.5

additional figures and experiments that help you illustrate your point and support your claims.

6   2K Analysis (90 pts)

For 3 client machines (with 64 total virtual clients per client VM) measure the throughput and response time of your system in a 2k experiment with repetitions. All GET operations have a single key. Investigate the following parameters:

- Memcached servers: 1 and 3

- Middlewares: 1 and 2

- Worker threads per MW: 8 and 32

Repeat the experiment for (a) a write-only and (b) a read-only workload. For each of the two workloads, what is the impact of these parameters on throughput, respectively response time?

Number of servers	1 and 3
Number of client machines	3
Instances of memtier per machine	1 (1 middleware) or 2 (2 middlewares)
Threads per memtier instance	2 (1 middleware) or 1 (2 middlewares)
Virtual clients per thread	32
Workload	Write-only and Read-only
Multi-Get behavior	N/A
Multi-Get size	N/A
Number of middlewares	1 and 2
Worker threads per middleware	8 and 32
Repetitions	3 or more (at least 1 minute each)

7   Queuing Model (90 pts)

Note that for queuing models it is enough to use the experimental results from the previous sections. It is, however, possible that the numbers you need are not only the ones in the figures we asked for, but also the internal measurements that you have obtained through instrumentation of your middleware.

Table 2: Write-Only

Factor	Throughput (ops/sec)				Response Time (ms)			
	Effect	Sum of Squares	Percentage of Variation	Confidence Interval 90%	Effect	Sum of Squares	Percentage of Variation	Confidence Interval 90%
I	8193	1611160k		$(8128, 8259)$	22.4	12027		$(22.2, 22.6)$
S	-919	20250k	18.9	$(-984, -853)$	2.3	125	12.3	$(2.1, 2.5)$
M	960	22116k	20.6	$(895, 1025)$	-3.2	247	24.2	$(-3.4, -3.0)$
W	1610	62181k	58.0	$(1544, 1675)$	-5.1	617	60.4	$(-5.3, -4.9)$
SM	-110	291k	0.3	$(-175, -45)$	-0.5	6	0.6	$(-0.7, -0.3)$
SW	-261	1633k	1.5	$(-326, -196)$	-0.2	1	0.1	$(-0.4, -0.0)$
MW	91	197k	0.2	$(25, 156)$	0.9	19	1.9	$(0.7, 1.1)$
SMW	-10	2k	0.0	$(-75, 55)^a$	-0.1	0	0.0	$(-0.3, 0.1)^a$
Error		535k	0.5			5.0	0.5	

Table 3: 2^k3 Experiment Base Table for Read-Only

i	I	S	M	W	SM	SW	MW	SMW	Throughput (ops/sec)		Response Time (ms)	
									(y_{i1}, y_{i2}, y_{i3})	$\hat{y}_i$	(y_{i1}, y_{i2}, y_{i3})	$\hat{y}_i$
1	1	-1	-1	-1	1	1	1	-1	(2890, 2895, 2898)	2894	(63.0, 63.1, 63.0)	63.1
2	1	1	-1	-1	-1	-1	1	1	(7779, 8234, 8203)	8072	(21.6, 21.5, 21.6)	21.6
3	1	-1	1	-1	-1	1	-1	1	(2879, 2790, 2879)	2849	(62.3, 60.1, 62.6)	61.7
4	1	1	1	-1	1	-1	-1	-1	(8339, 8610, 8481)	8477	(18.5, 20.1, 19.7)	19.5
5	1	-1	-1	1	1	-1	-1	1	(2903, 2905, 2896)	2902	(62.8, 61.8, 62.7)	62.4
6	1	1	-1	1	-1	1	-1	-1	(8693, 8686, 8693)	8690	(20.1, 19.6, 20.1)	19.9
7	1	-1	1	1	-1	-1	1	-1	(2885, 2877, 2868)	2877	(62.1, 61.6, 61.7)	61.8
8	1	1	1	1	1	1	1	1	(8635, 8621, 8613)	8623	(18.8, 19.1, 19.8)	19.2

Table 4: Read-Only

Factor	Throughput (ops/sec)				Response Time (ms)			
	Effect	Sum of Squares	Percentage of Variation	Confidence Interval 90%	Effect	Sum of Squares	Percentage of Variation	Confidence Interval 90%
I	5673	772404k		$(5636, 5710)$	41.1	40635		$(40.9, 41.4)$
S	2793	187162k	99.5	$(2756, 2829)$	-21.1	10685	99.8	$(-21.3, -20.9)$
M	33	27k	0.0	$(-4, 70)^a$	-0.6	9	0.1	$(-0.8, -0.4)$
W	100	240k	0.1	$(63, 137)$	-0.3	2	0.0	$(-0.5, -0.1)$
SM	51	62k	0.0	$(14, 88)$	-0.1	0	0.0	$(-0.3, 0.1)^a$
SW	91	200k	0.1	$(54, 128)$	-0.2	1	0.0	$(-0.4, 0.0)^a$
MW	-56	76k	0.0	$(-93, -19)$	0.3	2	0.0	$(0.0, 0.5)$
SMW	-62	91k	0.0	$(-98, -25)$	0.1	0	0.0	$(-0.1, 0.3)^a$
Error		172k	0.1			6.2	0.1	

7.1   M/M/1

Build queuing model based on Section 4 (write-only throughput) for each worker-thread config-uration of the middleware. Use one M/M/1 queue to model your entire system. Motivate your choice of input parameters to the model. Explain for which experiments the predictions of the model match and for which they do not.

7.2   M/M/m

Build an M/M/m model based on Section 4, where each middleware worker thread is represented as one service. Motivate your choice of input parameters to the model. Explain for which experiments the predictions of the model match and for which they do not.

7.3   Network of Queues

Based on Section 3, build a network of queues which simulates your system. Motivate the design of your network of queues and relate it wherever possible to a component of your system. Motivate your choice of input parameters for the different queues inside the network. Perform a detailed analysis of the utilization of each component and clearly state what the bottleneck of your system is. Explain for which experiments the predictions of the model match and for which they do not.